IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4886

A Native Text Database: What for?

Thomas B. Hodel-Widmer
University of Zurich, Department of Information Technology, Winterthurerstr. 190, CH-8057 Zurich, Switzerland
hodel@ifi.unizh.ch

Klaus R. Dittrich
University of Zurich, Department of Information Technology, Winterthurerstr. 190, CH-8057 Zurich, Switzerland
dittrich@ifi.unizh.ch

ABSTRACT

A significant gap lies between handling business (customer, product, finance, etc.) and text data (documents). Very often, word processing documents are stored somewhere within a confusing file structure with inscrutable hierarchy and low security. On the other hand, crucial data from an organization's point of view are stored in databases. The infrastructure and the data are highly secure, multi-user capable and available for several other tools to build reports, content and knowledge. Our idea is to use a similar philosophy for texts. Therefore, we strive for the storage of texts in a database in a native *way enables security and collaboration. By native, we mean that we can store text in a structured way in the database, so that database transactions can be applied.*

In this article, we describe our idea of turning text into valuable data. We present shortcomings of document processing, state of the art in document processing and a series of advantages for our database approach.

INTRODUCTION

Throughout the last few years, there have been a number of improvements in word processing applications including additional functionality, more intuitive interfaces and an integration with design, layout and other tools. While the basics of word processing applications remain the same, recent trends in organizations have meant a change in focus towards collaboration functionalities. We see an increasing number of business requests for security, multi-user capability, definable business processes within documents and reuse concepts for content and layout.

PROBLEM STATEMENT

Over the last decade, organizations have spent a lot of time finding solutions for accessing structured data stored in database systems. However, this data represents a fraction of all available corporate information. A far larger volume exists as text in documents. These valuable sources of information are often inaccessible and not managed effectively. As organizations embrace globally interconnected systems and as a result start to build networked team, using information from many often unstructured sources, the problem is aggravated.

Document handling by word processing applications is more complicated than it looks. We have never seen an organization, independent of size, which has a suitable overview and control of its documents. Teamwork is not supported by word processing applications because sophisticated collaboration functionalities are missing. Most organizations are somewhat disorganized concerning security and storage structures of these documents. The reality today is that organizations can warehouse, and analytically process a small amount of corporate data that comprises numbers and dates, but the rest that remains as text often goes untapped. Important text assets stay buried within the organization, and the latent information they carry remains obscure to decision makers.

MISSING CONCEPTS

Within documents, there is no accurate storage system, low access security and low classification, poor or absent history, no integrated version control, and no flexible undo function. Collaborative editing, data lineage, and workflow capability are missing.

Security

Storage - The first deficiency is the confusing and insecure storage situation – there is no integrated safe storage, authorized user access as well as backup and recovery capability.

Example: Word processing applications store their data in the way of files. There are mainly three storage possibilities. First, the program can store the file on the local drive. Second, the application can store the file on a file server. Third, the user can use a document management tool, which stores the file on a file server or as binary object in a database, and the tool itself creates and stores an index of the content from the document.

Locally stored documents are highly insecure, there is no automatic backup and physical access to the hardware is easy. While document management tools usually have at least a detailed access management system, keep track of changes and create a detailed log file, they are, however, complicated, and create a significant amount of overhead.

Separate classification from any part of the document - No access concept down to the individual character based on user and roles is available. It is conceivable that a certain user has no read access to characters, characters combinations and / or their position within the text.

Example: A project member can, at the very same moment, open the same document as his boss, but based on the access rights some parts may be hidden. The access concept can be used also for defining unchangeable text and is a part of the workflow concept.

Complete history over the document creation - The history of a document, from creation up to saving, especially during editing, is missing.

Example: This data includes time, date, and author, so that it is possible to reconstruct the exact creation process of the document. Transparent information, such as who changed what on which date and time over the development phase of the text will be available.

Integrated versioning - Versions must be an integrated part of the document.

Example: The versioning can be done automatically by the system, based on rules defined by the user, or it can be evoked manually. At any time, a user can retrieve every version of a text and navigate forward and backwards throughout the text creation.

Cases dependent undo function - A local or global undo function for any part of the document is needed. This undo function can also depend on user and roles, time and date, and on any combination of it.

Collaboration Functionalities

Simultaneous writing in a shared document. Several people must have the possibility to read, write and edit the same document simultaneously.

Example: Some functions of collaboration built into today's word processing applications are to compare document versions, merge documents and link parts of documents.

Collaborative business process. It should be possible to define a workflow within or for a document. It should also be possible to set up the user or role (which person is in charge) allowed to write, edit, sign, or review any part of a text and in which way (concerning time, serialized or parallelized) this is done.

Example: It is possible to define the person who has to release or electronically sign off a certain part, or the whole document. Roughly speaking all known workflow functions can be included.

Data lineage. In daily business, very often documents are based on other documents; most of the content remains the same. Hundreds of rather new documents often refer to some few core documents. There are always plenty of associations to other documents. Often a part of a text is used somewhere else, or a certain text is a kind of a response to another part of a document. These associations have to be known.

Example: A copy-and-paste part of the text should be able to identify its source roots. As soon as the original part is being edited, the system can offer an update of the 'copy-and-paste' function to the user.

Distributed Teams. All collaboration functionalities together should enable distributed (virtual) teams in the field of text creation.

Flexible handling of content and layout. Information about content and layout needs to be separated, so that a flexible presentation of the same document is made possible.

Example: On one side is the defined content and on the other side are several layout definitions which can be applied to the content. There will be no confusion between content and layout because the system never has to convert a document from one format to another.

File and locate documents. The way to file, manage, and locate documents has to be redesigned to reach a placeless document philosophy [Dourish 2000]. The users need functionality to store and locate documents without specifying a location, and without using a fixed hierarchy.

Example: Most users organize their documents by location in hierarchies onto which they map their own semantic structures. More generally expressed, hierarchies pervade document and information storage systems.

Automated multichannel publishing. It should be possible to publish all text documents in multiple forms.

Example: Multiple forms like web, printing, create PDF-, MS-Word- or a XML-file and others, while keeping the document consistent, up-to-date and complete.

Content and Knowledge Management

If the focus is on documents, today's content and knowledge management systems are limited to the included text. This means these tools store just the content, the text of the document and nothing else. Our proposal includes information about the creation process, such as who wrote what for which group, to which documents this specific document is associated, who has what kind of access to the document, which part is a copy from another document and so on. This is crucial information in creating content and knowledge out of word processing documents. Word processing documents hold crucial information and are therefore an important part of an organization's knowledge; a fact, which is prevailingly underestimated.

Examples: Four documents which cover relevant knowledge for an upcoming project were found in a company. Based on the information about which part was written from which author and which part was copied from another document, the system can find out suitable employees and teams to discuss the new project.

An other use would be for example, that the system finds similar sentences and paragraphs during editing a text.

CURRENT RESEARCH STATUS OF WORD PROCESSING FUNCTIONALITIES

The aforementioned troubles with the handling of text documents occur in nearly all organizations and in all teams, which have to work together supported by computers. Solutions in this area are astonishingly scene. We have found very few commercial solutions which support even one of the mentioned functionalities in the domain of security, collaboration and knowledge management.

Security

Security mechanisms on the document level belong to standard solutions. This is carried out using the file system or a document management environment. Security settings within a document exist, but are always applied to all users. We were not able to find a flexible system in which it is possible to set read, write and grants rights for any parts within a document for any group or a specific user.

Collaboration

Commercial tools in this field include 'Windows Messenger' with 'Application Sharing'[1] and 'Lotus Sametime'[2]. These applications offer the possibility to share any started program and work together within the same environment. Several users can control the shared program, but not at the same time, they must pass the control from one to the other. Interesting projects from universities are the following:

- DCWA provides group services, maintaining a unique version of a document, facilitating both organizational and semantic relations among parts of the document as well as an interface for the current working area and for users' own viewing spaces. It is a distributed synchronous and asynchronous collaborating environment [Chang 1995].
- REDUCE 'REal-time Distribiuted Unconstrained Collaborative Editing' supports concurrent editing of any text at any time. It also supports concurrent undoing of any operation at any time, instant response on distant editing over the internet, optimistic concurrency control by operational transformation, convergence, causality-preservation and intention-preservation, integrated and dedicated group-awareness support, optional and responsive locking support, web-based interface for the Document repository system and enables multiple concurrent collaborative sessions [Sun 1997]. REDUCE is the most sophisticated collaborative text editor we found.[3]
- NetEdit provides centralized file and session management, unconstrained group editing of documents, and chat session management [Zafer 2001].
- There are as well some other collaborative text editor applications, that support access by multiple simultaneous users, like GROVE [Ellis 1991], ShrEdit [Dourish 1992] and Jupiter [Nichols 1995].
- SubEthaEdit[4] is sleek collaborative editor. It supports some collaborative awareness functions and is mainly concentrated on software developers.

All the mentioned projects lack new functionalities as described above, except for simultaneous writing. Furthermore, not one of the current applications solves the security issues as described, and no project supports knowledge management issues. The focus of new word processing applications lies primarily and often only on collaboration functions.

Text Retrieval / Mining

An important question in the research field within text databases is how unstructured documents should be handled in order to make them searchable [Salminen 1987]. An algebra for structured office documents for filing, retrieval and construction of such objects was discussed as well [Güting 1989]. Computer linguistics developed and discussed the different text retrieval methods, like document, probabilistic, vector and passage retrieval [Kaszkiel 1999]. A very special and interesting attempt is the model for querying textual databases by contents and structure of the text [Navarro 1997]. Text retrieval, from simple search engines up to sophisticated text mining tools is the most researched field within text databases.

Document, Content and Knowledge Management Systems

These systems, like 'Autonomy'[5], 'Documentum'[6], 'FileNet'[7], 'OpenText'[8], 'SAS Text Miner'[9], and 'Thunderstone'[10], just to mention some of them, have a very similar philosophy. First, they import documents, which means that the system stores the document as it is in a file system (this is the common way), or within the database (this is the exception). Some of them even have both possibilities. Second, the tool analyzes the content of the document and creates a full text index. This index is normally stored within the file system. Both, document and index are stored in a file system, for performance reasons. If the tool is more sophisticated and supports knowledge management and text mining, automatic or semi-automatic categorization is possible. Truncating, stemming or lemmatization and morphological analysis are provided, too. If document management is supported, automatic versioning from the whole document is the standard as well as check in and check out functionalities; also a certain kind of access control is included. Access rules as well as locking a document for editing is always applied to the document as a whole.

Support from Databases

Standard support - In general, all database systems support the storage of plain text within the database. The most common and important data types are, among others, character, variable character, long, large objects, character large objects, binary large objects, and binary file.

Enhanced support - A text system that indexes any document or textual content to deliver fast and accurate retrieval of information is more or less offered by commercial database systems. Oracle for instance introduced 'Oracle Text', and IBM calls its similar product 'DB2 Text Extender', and 'DB2 Text Information Extender'. In the following we summarize the supported functionalities of these systems.

All mentioned text systems offer a complete text search solution. Such a system provides specialized text indices for traditional full text retrieval applications for documents, document classification, text warehousing, document libraries and archives. It excels at performing exact and inexact matches, word positioning comparisons, intelligent match, high-accuracy relevance ranking of returned results, and XML searches. These systems can filter and extract content from all commonly used document formats. Moreover, they offer a set of multilingual features, data partitioning, query optimization to ensure the best response time, not only for pure text queries, but also for 'mixed' queries that combine text search with database search, integrated security to protect all information assets with the same rigor as your database data, however.

To improve search quality, these systems use advanced features like thesauri which consist of a controlled vocabulary with a structure that denotes hierarchy and relationships among the words as the base for the linguistic engine that can analyze and generate the main themes of a piece of text. These types of features are extremely useful for building classification for incoming sets of documents based on their content.

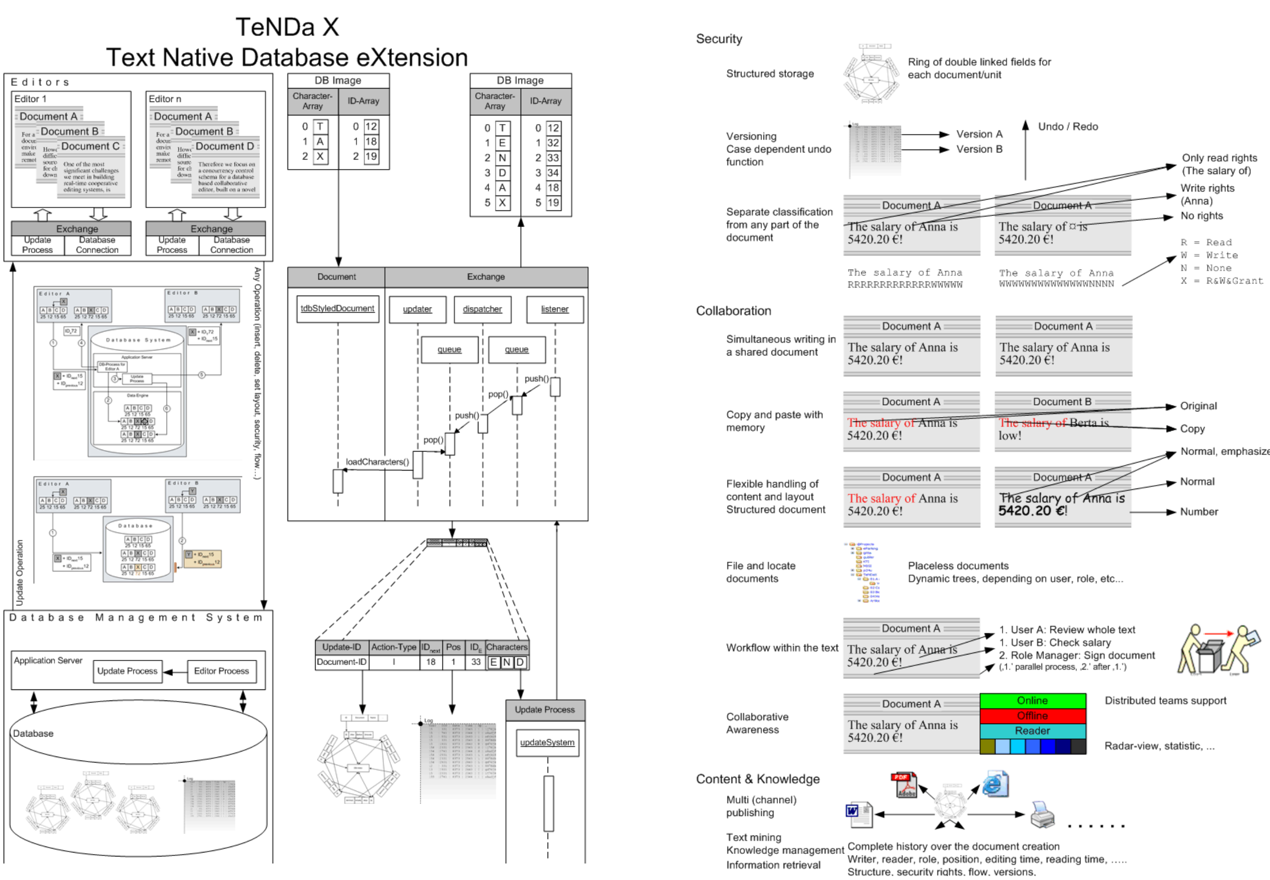
DATABASE APPROACH

To improve upon the current situation and to allow the missing concepts defined in part 3 to become reality, different approaches are possible. From our point of view, however, there is only one promising approach to improving this situation and arriving at an open system for further innovative ideas. We call it a database based word processing concept (see Figure 1). Historically, text has been perceived as requiring a different set of technologies for retrieval and management than structured data. This perception has not only burdened organizations with multiple storage systems and development environments but has also stood in the way of effectively integrating all organization information assets into a database.

Native Text Database

We are convinced that word processing applications should store data in a 'native' way in a database and then benefit from the advantages of a database management system like querying the content, restricting access, persistent storage, inference and rule-based actions, multiple user

*Figure 1: **Database based word processing concept for the realization of the mentioned functionalities***



interfaces, representing complex relationships among data, integrity constraints, backup and recovery, and much more. Furthermore, it is possible to build a set of transactions for dealing with text, and to build a standardized API language so that different database systems as well as different front-end applications could be used. Another important argument is that the appropriate handling of database servers is well known in organizations. These aspects of integration are also greatly beneficial to companies as they would not have to undergo a paradigm shift to learn to manage their text assets. According to this know-how, one part of the security issues will automatically be solved by existing technology. Databases are the best foundation for text mining and content and knowledge management. Existing tools can get direct access to the 'native text database management system' to get a complete history of the document creation, authors and readers, access history, access roles and security. A new range of crucial data for text mining and content and knowledge management will be available.

In summary, the database approach offers a variety of interesting solutions in the fields of security, collaboration, text mining, and content and knowledge management.

In the 'Lowell Database Research Self-Assessment' report [Gray 2003], where senior database researchers gathered to assess the state of database research and to define problems and problem areas that deserve additional attention, an integration of text, data, code, and streams was recommended, as seen in the first point within this report. This is exactly what we are trying to do. Up to now, the DBMS field focused "on capturing, organizing, storing analyzing, and retrieving structured data." The TeNDaX approach tries to extend DBMS to also manage text. This addition must be made 'clean' and the responding sophisticated data type should appear as a 'first-class citizen' of a DBMS (like integers, character strings, etc.). Based on this structure, collaborative business processes can be applied. Furthermore, the query language has to be elaborated upon with functions that operate on these extended data types.

CONCLUSION

Data handling of word processing applications is based on proprietary technology that prevents further innovative development of any kind. The implementation of the newly needed functionalities, stemming from business requests on how to create documents and use their content, demands a redesign.

Therefore, we constructed a Text Native Database eXtension. TeNDaX[1] is an experimental system that we are using to explore the issues of database management system that is organized entirely around text. TeNDaX is not, in itself, a database; instead, it provides uniform coordinated access to a native text extension from current database management system. TeNDaX provides the means for editing a document in terms of transactions.

Concept, prototype (see Figure 2), transaction and performance evaluation of the mentioned ideas in this article are described in [Hodel 2003].

Figure 2: TeNDaX Prototype

ENDNOTES

1. http://www.microsoft.com/windowsxp/windowsmessenger/
2. http://www.lotus.com/products/lotussametime.nsf/wdocs/homepage
3. http://reduce.qpsf.edu.au/index.html
4. http://www.codingmonkeys.de/subethaedit/
5. http://www.autonomy.com/
6. http://www.documentum.com/
7. http://www.filenet.com/
8. http://www.opentext.com/
9. http://www.sas.com/products/textminer/index.html
10. http://www.thunderstone.com/texis/site/pages/Home.html
11. http://www.tendax.net

REFERENCES

Chang, K.H. et al.: On computer supported collaborative writing tools for distributed environments. In: Proceedings of ACM, Computer science, 1995, 222 – 229.

Dourish, P. et al.: Awareness and Coordination in Shared Workspaces. In: Proceedings of ACM, CSCW'92, 1992, 107–114.

Dourish, P. et al.: A programming model for active documents. In: Proceedings of ACM, User interface software and technology,2000, 41 – 50.

Ellis, C.A. et al.: Groupware: Some Issues and Experiences. In: Communications of the ACM 34, 1991, 38–58.

Gray, J. et al.: The Lowell Database Research Self Assessment. Lowell Massachusetts, 2003. (http://research.microsoft.com/~Gray/Lowell/)

Güting, R.H. et al.: An algebra for structured office documents. In: Proceedings of ACM, Transactions on Information Systems, Vol. 7, No. 2, 1989, 123 – 157.

Hodel, T.B., Technical Report University of Zurich, TR 12003101, Zurich 2003. (http://www.tendax.net)

Kaszkiel, M. et al.: Efficient passage ranking for document databases. In: Proceedings of ACM, Transactions on Information Systems, Vol. 17, No. 4, 1999, 406 – 439.

Navarro, G. et al.: A Model to Query Document Databases by Content and Structure. ACM Press, 1997, 400 – 435.

Nichols, D.A. et al.: High-Latency, Low-Bandwidth Windowing in the Jupiter Collaboration System. In: Proceedings of ACM, UIST'95, 1995, 111–120.

Salminen, A. et al.: A Relational Model for Unstructured Documents. In: Proceedings of ACM, SIGIR, 1987, 196 – 207.

Sun, C. et al.: REDUCE: a prototypical cooperative editing system. In: Proceedings of ACM, Human-Computer Interaction, 1997, 89 – 92.

Zaffer, A. et al.: A Collaborative Editor. Technical Report TR-1-13, Computer Science, Virginia Tech, 2001.